

Trabalho Prático de Engenharia de Software

Evento Esportivo

Celso Junio Simões
Felipe Matheus Guimarães
Matheus Ferreira Coelho

Professor: Eduardo Figueiredo

Grupo 7

Sistema para acompanhamento de evento esportivo

- **O objetivo do sistema é fornecer informações sobre um evento esportivo (exemplo: olimpíada, copa do mundo, etc.).**

Evento: Copa do Mundo



Evento: Copa do Mundo

Informações gerais:

- Local: Catar;
- Data: 21/11/2022 até 18/12/2022;
- Horários de Jogos: 07:00; 12:00; 16:00;
- Estádios: Lusail, Ras Abu Aboud, Khalifa, Al Bayt...
- Equipes: Serão decididas pelas eliminatórias
- Sedes: Doha, Al Khor, Al Rayyan, Al Wakrah e Lusail

Requisitos: Evento Esportivo

- Dentre as informações, deve ser possível acompanhar a tabela de jogos com informações sobre local, data e horário (dentre outras).
- O usuário também pode visualizar informações específicas sobre cada sede ou praça esportiva. Por exemplo, para cada sede, o sistema deve apresentar informações sobre o estádio, aeroporto(s) próximo(s) e hotéis.

Requisitos: Evento Esportivo

- O sistema também deve fornecer informações sobre as equipes participantes do evento. Além disso, o usuário do sistema pode ver os cruzamentos futuros das equipes que se classificarem.
- O evento esportivo escolhido pelo grupo foi a Copa do Mundo, pois a próxima edição já será em 2022 e achamos interessante e viável para o nosso trabalho.

Modelo de Processo

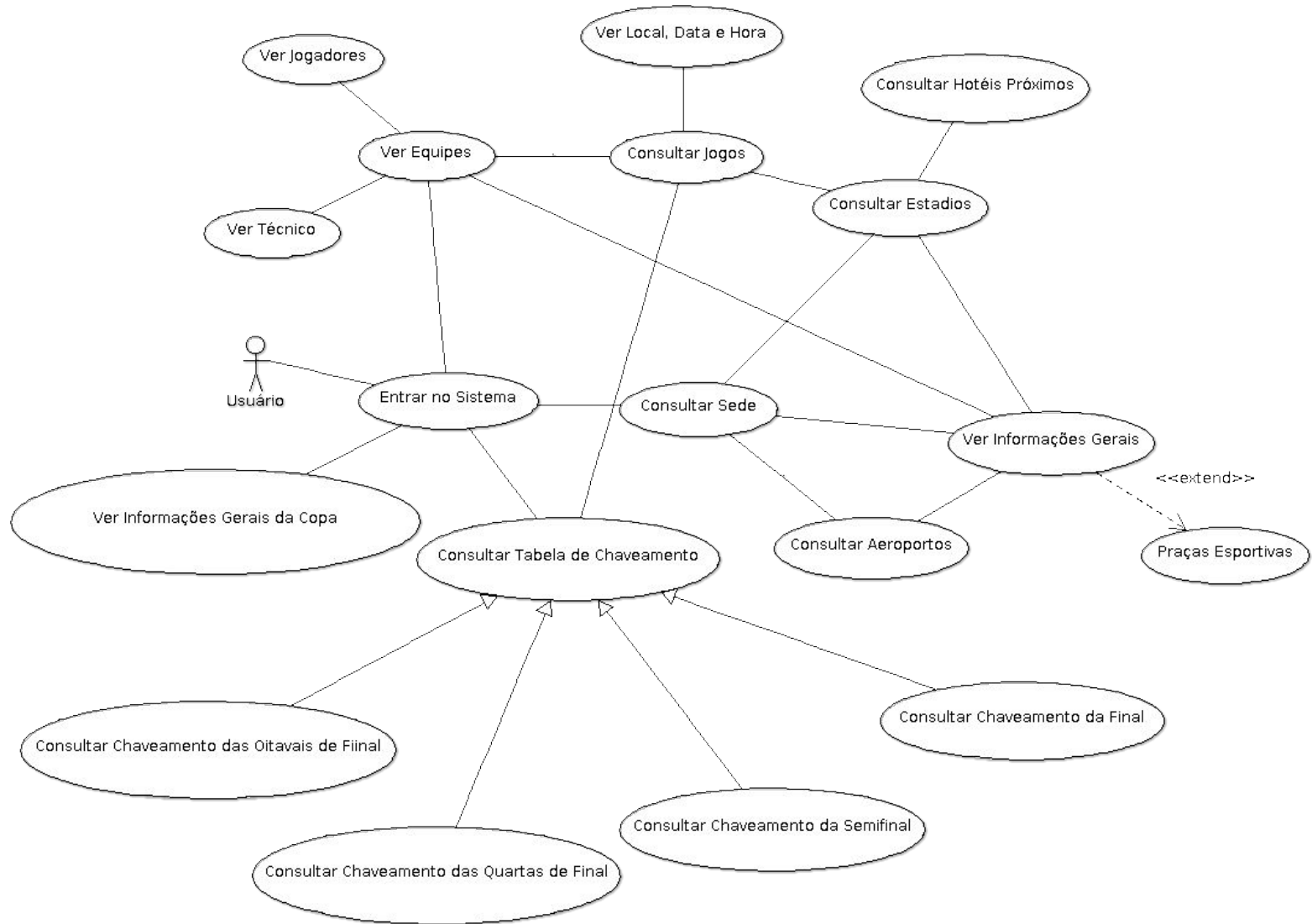
- O modelo de processo escolhido foi o Scrum.
- A board com os requisitos e tasks está disponível neste link do Trello:
<https://trello.com/b/vVcqiknk/tp-de-eng-software>

Algumas das tasks

[Task 4] Criar classe de Equipes

- Criar o classe de Ver Equipes:
 - Se o Usuário vier da Home, dar opções de retorno e selecionar método;
 - Adicionar ao fluxo

Diagrama de Casos de Uso



Alguns dos cenários

Cenário 3:

- Nome do Cenário: Ver equipes
- Atores:
 - Usuário
- Pré-condição:
 - Entrar no sistema
- Fluxo normal
 - Verificar as equipes que se classificaram para jogar a Copa do Mundo
- Fluxos alternativo:
 - Nenhum
- Pós-condição:
 - O usuário volta ao início, vê os jogos que essa equipe jogou, vê os jogadores de determinado time, vê as informações do técnico ou vê as informações gerais da equipe.

Alguns dos cenários

Cenário 14:

- Nome do Cenário : Consultar tabela de chaveamento
- Atores:
 - Usuário
- Pré-condição:
 - Estar dentro do sistema
- Fluxo normal
 - Entrar no sistema
 - Selecionar para consultar a tabela de chaveamento
- Fluxos alternativo:
 - A copa ainda se encontra na fase de grupos e o chaveamento não está disponível
- Pós-condição:
 - O chaveamento do mata-mata é mostrado ao usuário

Implementação: Decisões

Processo:

Para possibilitar a facilidade de alinhamento entre o time, utilizamos da ferramenta 'Code With Me' para codarmos juntos, no mesmo projeto, ao mesmo tempo.

Código:

O Código fora inteiramente escrito em Java, como solicitado, sem fazer uso de nenhuma biblioteca externa da linguagem, somente nativas:

```
src > Home.java
1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.Locale;
4  import java.util.Scanner;
```

```
src > Jogo.java
1  import java.time.LocalDate;
```

Além disso, o programa foi terminado com sucesso, estando atualmente em seu estado funcional.

Implementação: Organização e Arquitetura

Requisitos:

Vide Slides 5 e 6.

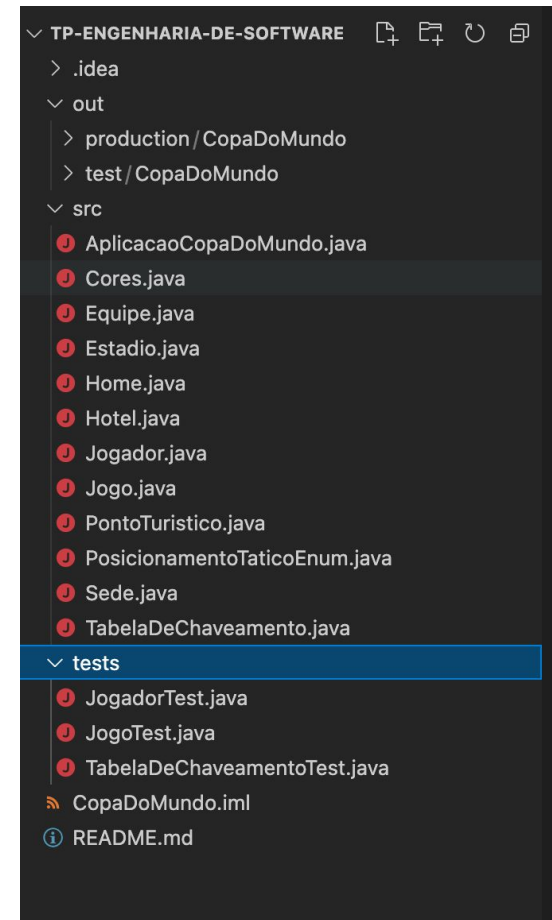
Organização de arquivos:

Detalhamos as classes, métodos e atributos mais importantes através de comentários, possibilitando a melhor compreensão do código por qualquer pessoa

Componentes principais:

- Home.java; - Agrega todos os fluxos
- PontoTuristico.java; --Herança

Utilizamos a idéia do método de Scrum, de tasks, enquanto utilizamos uma aproximação da Arquitetura de Repositório para manusear o código em si
Já que, em todas as vezes, codamos o projeto ao mesmo tempo, no mesmo arquivo



Exemplo de código

[illegible]

Exemplo de teste: JogoTest

Decidimos fazer testes de componentes, para abranger melhor o projeto e código desenvolvido, tendo em vista que a maioria dos nossos testes unitários se tratam de gets e sets.

```
JogoTest.java ×
tests > JogoTest.java
1  import org.junit.jupiter.api.Test;
2
3  import static org.junit.jupiter.api.Assertions.*;
4
5  public class JogoTest {
6
7      @Test
8      public void itShouldVerifyTheWinner(){
9          Equipe brasil = new Equipe("Brasil", "Tite");
10         Equipe argentina = new Equipe("Argentina", "Lionel Scaloni");
11
12         Jogo brasilVsArgentina = new Jogo(brasil, argentina, 3, 2);
13
14         assertEquals(brasilVsArgentina.getVencedor().getNome(), brasil.getNome());
15     }
16
17     @Test
18     public void itShouldReturnNullIfItIsATie(){
19         Equipe brasil = new Equipe("Brasil", "Tite");
20         Equipe argentina = new Equipe("Argentina", "Lionel Scaloni");
21
22         Jogo brasilVsArgentina = new Jogo(brasil, argentina, 3, 3);
23
24         assertNull(brasilVsArgentina.getVencedor());
25     }
26
27 }
```

Exemplo de teste: TabelaDeChaveamento

TabelaDeChaveamentoTest.java X

tests > TabelaDeChaveamentoTest.java

```
1  import org.junit.jupiter.api.Test;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  import static org.junit.jupiter.api.Assertions.*;
7
8  public class TabelaDeChaveamentoTest {
9      @Test
10     public void itShouldValidateTheSemifinalWinners() {
11         TabelaDeChaveamento tabela = new TabelaDeChaveamento();
12
13         List<Jogo> semi = new ArrayList<>();
14         Equipe alemanha = new Equipe("Alemanha", "Hans-Dieter Flick");
15         Equipe argentina = new Equipe("Argentina", "Lionel Scaloni");
16         Equipe brasil = new Equipe("Brasil", "Tite");
17         Equipe franca = new Equipe("França", "Didier Deschamps");
18
19         Jogo jogo1 = new Jogo(alemanha, franca, 3, 2);
20         Jogo jogo2 = new Jogo(brasil, argentina, 5, 0);
21
22         semi.add(jogo1);
23         semi.add(jogo2);
24         tabela.setSemifinal(semi);
25
26         assertEquals(tabela.getSemifinal().get(0).getVencedor().getNome(), alemanha.getNome());
27         assertEquals(tabela.getSemifinal().get(1).getVencedor().getNome(), brasil.getNome());
28     }
29 }
```


Diagrama de Classes

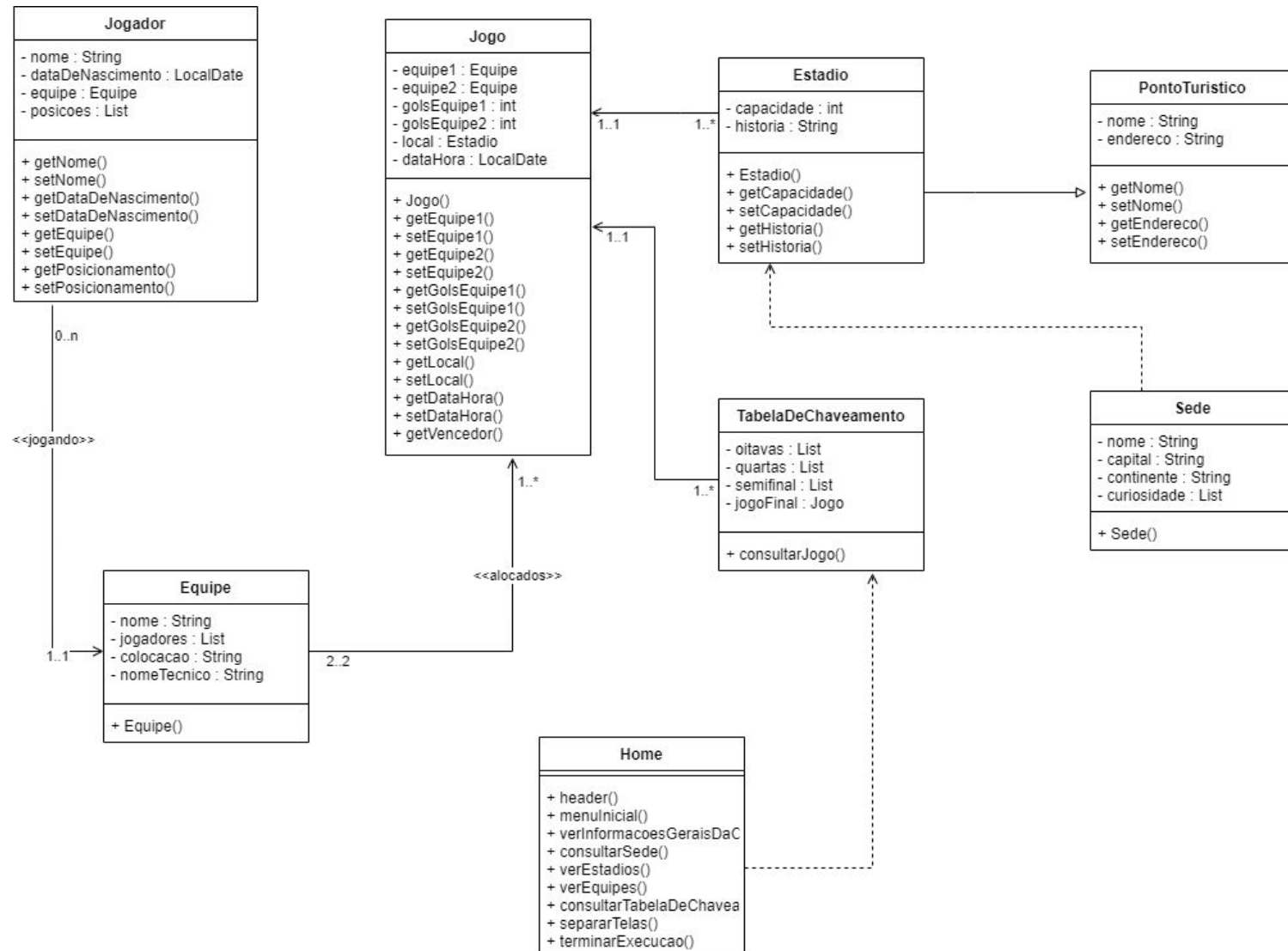


Diagrama de Sequência

Usuário

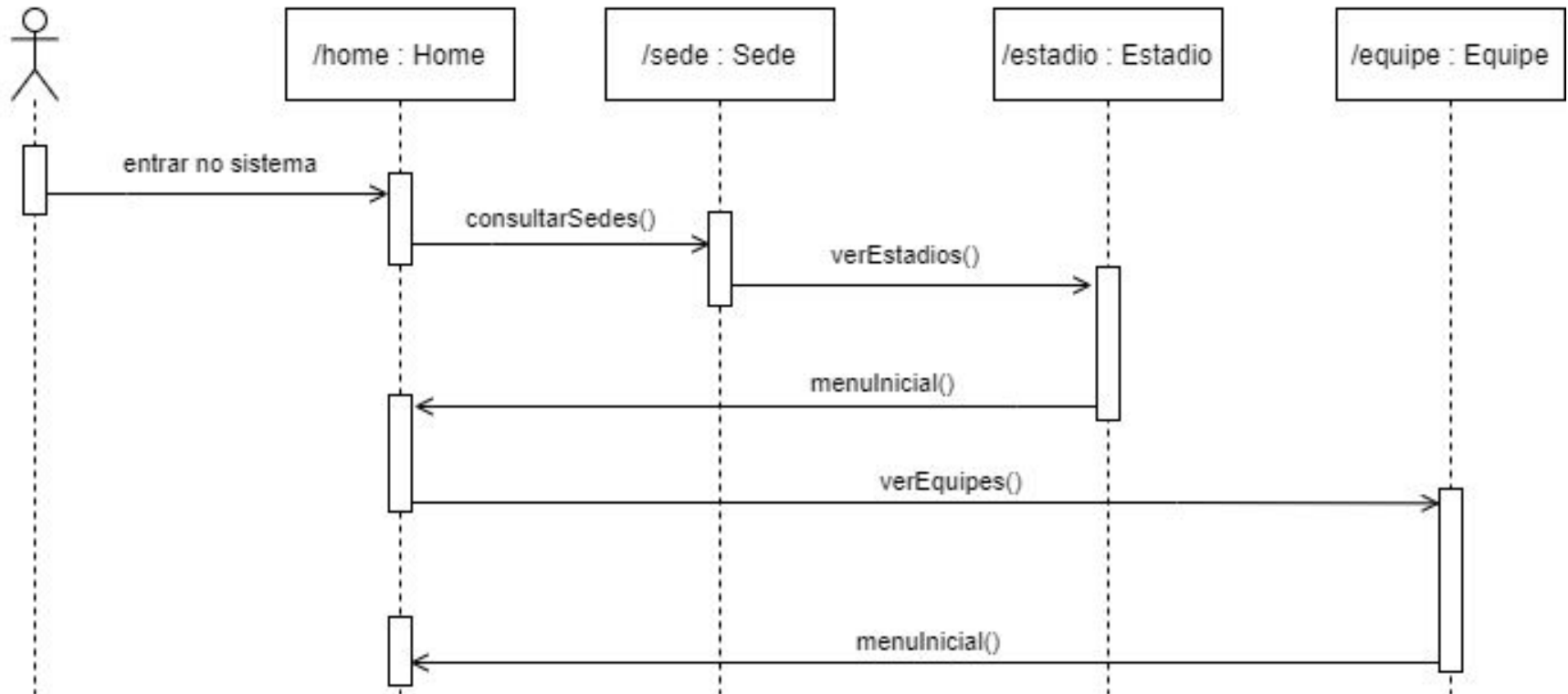
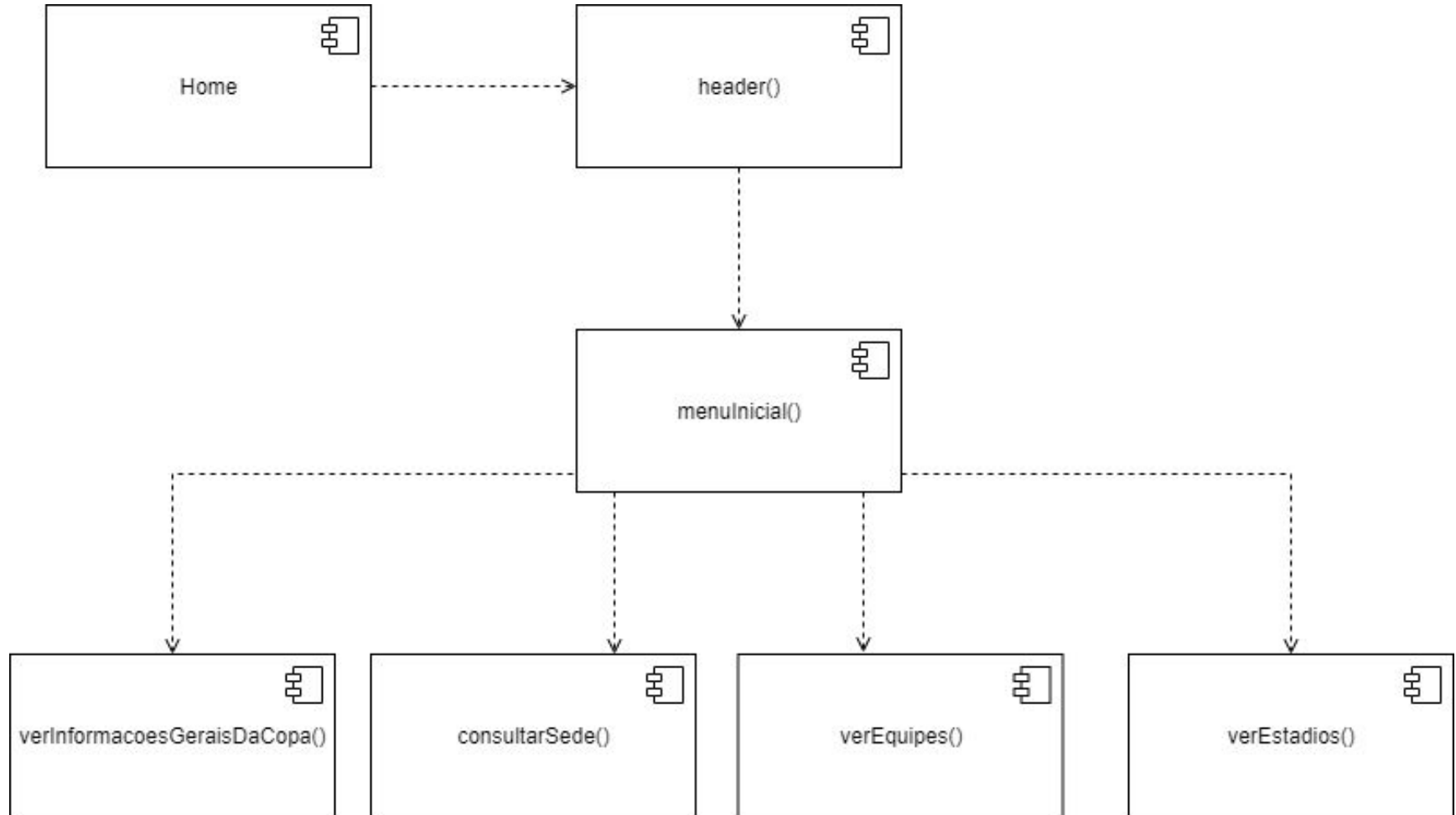


Diagrama de Componentes



Execução do Sistema

(~Felipe Guimarães)





MUITO
OBRIGADO!